Dress Code: A Virtual Fashion Assistant Powered by Large Language Models

Gabby Cordelli, Disha Dubey, Anjali Goel, John Hope
University of Virginia
School of Data Science
{gec2tp, fnj7ht, arg8qqv, jah9kqn}@virginia.edu

Abstract

In an age where artificial intelligence continues to reshape industries, fashion remains a uniquely personal and expressive domain. This project explores the design and deployment of "Dress Code," a virtual fashion stylist powered by a large language model (LLM). Our system generates personalized outfit recommendations tailored to users' existing wardrobes, local weather conditions, and aesthetic preferences.

We leverage an OpenAI GPT-4o Mini model accessed via OpenAI API, integrating it with a Chromadb database populated by user wardrobe inputs. External contextual data is sourced from the Open-Meteo Weather API and Pinterest Trends. User clothing images are first processed via computer vision models to generate embeddings and captions, supporting natural language querying of existing wardrobes.

Experiments across multiple user test cases demonstrate the model's ability to retrieve contextually appropriate outfits that blend user-provided items with real-time environmental and trend data. Rather than relying on quantitative metrics, we evaluate outputs qualitatively based on readability, relevance, and correctness.

Our findings show that LLMs, when combined with external APIs and computer vision techniques, can deliver coherent, trend-aware styling advice. This work highlights the promise of LLM-driven virtual assistants for practical, real-world personalization tasks, and lays a foundation for future expansion into social features, retailer integrations, and broader style discovery.

1 Introduction

In an age where artificial intelligence continues to reshape industries, fashion remains a uniquely personal and expressive domain—one that stands to benefit from the thoughtful integration of generative technologies. This project explores the design and deployment of a virtual fashion stylist powered by a large language model (LLM), aimed at helping users generate outfit recommendations tailored to their wardrobe, preferences, and local weather conditions.

While our initial vision included expansive integrations such as Pinterest boards and retailer data, we have since refined our scope to prioritize technical feasibility and user-centered design. Our final system leverages the OpenAI GPT-4o Mini model, hosted on Hugging Face, chosen for its strong instruction-following capabilities and efficiency in edge deployment. The model is paired with the Fashion MNIST [1] dataset for garment recognition, Pinterest trends data[2], and an integrated APIs for contextual inputs (local weather data).

Users can upload images or provide descriptions of their wardrobe, specify preferences (e.g., occasion, comfort level), and receive customized outfit suggestions through a conversational interface. This paper outlines our system architecture, model design, data pipeline, and implementation strategy for delivering an intelligent,

*Course Project for DS-6051 – Decoding Large Language Models.

trend-aware virtual styling experience.

2 Related works

Rule-based Stylist Systems. Early work in virtual fashion recommendation relied on structured rules and expert-defined logic. Wang and Rasheed's *Style-Me* system [3] exemplifies this approach by combining a fashion personality quiz with a rule-based engine to recommend outfits based on predefined compatibility rules. Though limited in adaptability, it was one of the first systems to integrate a basic learning mechanism for feedback-based personalization.

LLMs for Fashion Recommendation. Recent approaches have turned to large language models (LLMs) for more dynamic and context-aware outfit generation. Forouzandehmehr et al. [4] fine-tune a transformer-based LLM on the Polyvore dataset, using a vision-language model to extract attributes from clothing images and feeding them into the LLM for outfit composition. Their model incorporates user feedback, including negative samples, to improve style compatibility and trend awareness—closely aligning with our own approach using the GPT-4o Mini model.

Applications of Fashion-MNIST. While Fashion-MNIST is often used for benchmarking, some works have extended it toward real-world fashion tasks. Abd Alaziz et al. [5] validate their CNN and Vision Transformer models on Fashion-MNIST before transferring them to higher-resolution fashion product datasets. They build a visual similarity-based recommender system and demonstrate the limits of Fashion-MNIST in realistic scenarios, highlighting the need for richer image data in production systems.

LLMs for General Recommendation. Gao et al. introduce Chat-REC [6], a framework that prompts GPT-3.5 with user preferences and candidate items to generate personalized recommendations and natural-language explanations. Their system excels in cold-start and zero-shot scenarios, offering conversational and explainable outputs. While not fashion-specific, their methodology provides valuable insights for building stylist agents that interact fluidly with users and justify their suggestions.

3 Methodology

3.1 Data Sources

3.1.1 Fashion MNIST

The Fashion MNIST [1] dataset contains 70,000 grayscale images (28x28 pixels) of 10 clothing categories, with 60,000 for training and 10,000 for testing. These images are sources from Zalando, a German clothing retailer.

3.1.2 Open-Meteo Weather API

This is a free and open source weather API [7] that provides historical and forecast weather data using latitude and longitude as inputs.

3.1.3 Pinterest Trends

This website provides free access to historical and real-time Pinterest search data[2], allowing the analysis of keyword popularity across regions, time ranges, and demographics.

3.2 Model

3.2.1 OpenAI GPT-4o Mini

We use the OpenAI GPT-4o Mini model, a highly efficient instruction-tuned large language model accessed via the OpenAI API. GPT-4o Mini offers strong conversational abilities, contextual reasoning, and fast inference speeds, making it well-suited for integrating diverse contextual inputs (weather data, wardrobe embeddings, trend data) and generating personalized, context-aware fashion recommendations. Unlike open-weight models hosted locally, our system relies on API access to the GPT-4o Mini model for inference, ensuring up-to-date model capabilities and scalability.

3.3 Approach

To generate personalized fashion recommendations, we employ a Retrieval-Augmented Generation (RAG) architecture.

The first step in the process is the user inputting their location and stylistic preferences. Location is used to query our weather API to have the daily forecast, and stylistic preferences is used in our AI search and final prompting.

Next, users input images of their individual wardrobe items. We leverage a BLIP-2 model, which can generate natural language from images and prompts, to pull information about the items, including descriptions, colors, appropriate occasions and seasons, style, and gender. After generating these descriptions, we pass the text through a CLIP model to get text embeddings, which are then stored in a ChromaDB vector database collection specific to that user.

Then, when a user inputs a query to the system (e.g. "Going to a wedding"), we leverage several LLM agents to set up our AI search (context retrieval) to break down the available information into searchable phrases. The AI search is conducted using a weighted cosine similarity. The cosine similarity is heavily weighted for weather, as this is assumed to be the most important consideration, but also considers occasion, formality, and stylistic preference. The top 5 items within each category (top, bottom, jacket, shoes, other) are returned and stored as context. In instances where none of a category was returned, the search was conducted against items found in the FashionMNST dataset.

The final step is prompting our LLM, with all the given context, to provide an outfit recommendation. We provide the LLM the user's query, the summarized weather report, their stylistic preference, and the retrieved wardrobe items. A line at the end of the prompt includes the Pinterest trends and is given as consideration. From here, the LLM outputs its recommendations and file paths to the items it selected.

3.4 Experimental Setup

In order to test the efficacy of our LLM Fashion Stylist, we ran a test case using selected clothing items from each group member to be input into our Chromadb database. Our four test case users—Gabby Cordelli, Anjali Goel, Disha Dubey and John Hope—input about 30 png clothing files of their choice to be loaded into their own collections. These are meant to mimic the wardrobes users will input into the stylist to recommend outfit combinations. Importantly, our stylist makes recommendations based on users' existing wardrobes.

4 Experiments and Results

4.1 Experiment

Each user test case puts in different contextual query strings, locations, and desired styles in order to test the model's performance across different situations. Due to the qualitative nature of chatbot models, we do not judge performance on traditional quantitative markers such as loss and F1 score. Rather, we evaluate based on readability, relevance, and correctness of textual output.

4.2 Results

Our LLM stylist successfully identified user clothing, pulled in contextual details from our APIs, and integrated these details with the user's input specifications.

Here is an example input and output sample:

Input Sample:

Query: "I'm going for a long walk through the park."
Location: Charlottesville, VA
User: John
Aesthetic: Vintage

Output Sample:

- **Shirt:** Striped Tee (Brown)
  Reason: The striped tee is lightweight and breathable, making it perfect for warm

weather. Its casual style aligns with the event formality and adds a vintage touch.

- **Pants:** Sassy Joggers (Brown)
  Reason: Joggers are comfortable for walking and provide a relaxed fit. The brown color complements the tee and maintains a cohesive, vintage-inspired look.

- **Shoes:** Nike Swoosh (Green)
  Reason: These shoes are sporty and comfortable for walking. The green adds a fresh pop of color that works well with the brown and maintains a casual vibe.

- **Jacket:** Blue Jacket with Patch
  Reason: While the weather is warm, having a light jacket on hand is useful for any cooler breezes. The blue color is classic and fits the vintage style, while the patch detail adds a trendy touch.

As seen above, the model appropriately incorporates location-specific weather, occasion, and aesthetic into the output. The model successfully and smoothly incorporated this information, along with users' wardrobe data, into concise and helpful responses.

5 Discussion

Our experiments show that integrating a large language model with contextual external data sources can produce high-quality, personalized fashion recommendations. Across our test cases, the system successfully retrieved weather data, incorporated user wardrobe inputs, and adapted suggestions based on the specified event and aesthetic preferences.

A key strength of our approach was the seamless blending of diverse modalities—user wardrobe embeddings, weather APIs, and real-time trend data—into coherent, contextually appropriate outfit suggestions. The model outputs were consistently readable, relevant, and matched user intent, highlighting the power of LLMs in solving personalized, real-world tasks.

However, several challenges surfaced during development. The limited size and quality of the wardrobe inputs sometimes constrained the diversity of recommendations. Additionally, while Fashion MNIST provided a useful starting point for clothing classification, its simplicity and grayscale nature limited the stylist's ability to distinguish finer style details. Another limitation was our inability to access the Pinterest API directly; instead, we manually pulled a subset of trends, reducing the system's real-time adaptability.

Finally, the qualitative nature of chatbot evaluation remains a limitation—standard quantitative metrics like accuracy or F1 score are not easily applicable. As such, user satisfaction and subjective feedback would be crucial for more rigorous future evaluation.

Overall, this project demonstrates the feasibility of building a multi-source, LLM-powered virtual fashion assistant, while highlighting several opportunities for technical refinement and expansion.

6 Conclusion and Future Work

6.1 Conclusion

This project demonstrates the potential of large language models to deliver a personalized, context-aware virtual fashion assistant. By combining computer vision techniques for wardrobe cataloging, real-time weather data, and trend analysis, we created a system capable of producing practical and stylish outfit recommendations tailored to user preferences and external conditions.

Our experiments highlight the effectiveness of multi-modal integration and showcase how LLMs can bridge diverse data sources into coherent, human-centered outputs. While some limitations persist—such as wardrobe size constraints and qualitative evaluation metrics—the results are a strong indicator of the promise this approach holds for future intelligent styling applications.

6.2 Future Work

Several directions could further enhance our virtual stylist:

- **Social Features:** Integrating social functionality, such as sharing outfits with friends, collaborative closets, and outfit feedback, could make the experience more engaging and community-driven.
- **Retailer Integration:** Incorporating real-time retailer APIs would allow users to discover new clothing options to complement their existing wardrobe, complete with pricing and purchase links.
- **Discovery Mode:** A recommendation mode that suggests outfits outside the user's usual style preferences could help users explore new aesthetics and fashion trends.
- **Expanded Wardrobe Recognition:** Moving beyond Fashion MNIST to higher-resolution, color-rich datasets would improve the stylist's ability to differentiate between subtle garment details and offer more refined suggestions.

With these enhancements, the virtual stylist could evolve into a truly intelligent, trend-aware, and socially connected fashion assistant.

References

[1] Z. Research, "Fashion-MNIST Dataset on GitHub." `https://github.com/zalandoresearch/fashion-mnist`. Accessed: 2025-04-18.

[2] Pinterest, "Pinterest Trends." `https://trends.pinterest.com/`. Accessed: 2025-04-18.

[3] X. Wang and K. Rasheed, "Styleme: An intelligent fashion assistant system," in *2014 ASE/IEEE International Conference on Social Computing*, pp. 360–365, IEEE, 2014.

[4] N. Forouzandehmehr, N. Farrokhsiar, R. Giahi, E. Korpeoglu, and K. Achan, "Decoding style: Efficient fine-tuning of llms for image-guided outfit recommendation with preference feedback," *arXiv preprint arXiv:2409.12150*, 2024.

[5] H. M. Abd Alaziz, H. Elmannai, H. Saleh, M. Hadjouni, A. M. Anter, A. Koura, and M. Kayed, "Enhancing fashion classification with vision transformer (vit) and developing recommendation fashion systems using dinov2," *Electronics*, vol. 12, no. 20, p. 4263, 2023.

[6] Y. Gao, Z. Li, Y. Zhang, *et al.*, "Chat-rec: Towards interactive and explainable llms-augmented recommender systems," *arXiv preprint arXiv:2309.14476*, 2023.

[7] Open-Meteo, "Open-Meteo Weather API." `https://open-meteo.com/`. Accessed: 2025-04-18.